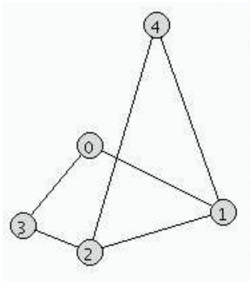
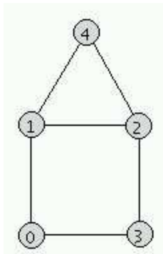
# 1 Layout von Graphen: Force-Directed Methoden

Force-Directed Methoden benutzen die Analogie zur Physik, um die Knoten eines Graphen zu positionieren. Wir betrachten einen ungerichteten Graphen $G = (V, E)$ als ein System von Körpern ($V$), auf die Kräfte ($E$) einwirken. Der Layout-Algorithmus sucht nach einer Konfiguration, in der die Gesamtsumme der Kräfte minimiert wird, d.h. er ordnet jedem Knoten eine Position zu, so dass die Kräfte, die auf jeden Knoten einwirken gleich 0 sind.

Im Allgemeinen bestehen force-directed Layout-Algorithmen aus zwei Teilen:

- einem Modell und
- dem Algorithmus




# 2 Spring-Layout

Beim Spring-Layout verwendet man als Model eine Kombination aus Federn und elektrischen Kräften. Die Kanten werden als Federn und die Knoten als sich abstoßend geladene Körper modelliert.

Sei $p_v = (x_v, y_v)$ der Vektor zum Knoten $v \in V$. Mit $||p_v - p_u||$ bezeichnen wir die Länge des Differenzvektor $p_v - p_u$ (der euklidische Abstand der Punkte $(x_v, y_v)$ und $(x_u, y_u)^1$). Desweiteren bezeichnet $\overrightarrow{p_u p_v}$ den Einheitsvektor $\frac{p_v - p_u}{||p_v - p_u||}$, der von $p_u$ nach $p_v$ zeigt.

Die abstoßende Kraft folgt der Gleichung[2]

$$f_0(p_u, p_v) = \frac{c_c}{||pv - pu||^2} * \overrightarrow{p_u p_v}$$

wobei $c_c$ eine Konstante ist, die angibt, wie stark sich zwei Knoten abstoßen.

Die Federkraft läßt sich durch die Gleichung[3]

$$f_1(p_u, p_v) = -c_s(||p_v - p_u|| - l_{uv}) * \overrightarrow{p_u p_v}$$

[1]$= \sqrt{(p_{v_x} - p_{u_x})^2 + (p_{v_y} - p_{u_y})^2}$

[2]Coulombsches Gesetz: $\vec{F} = \frac{1}{4\pi\epsilon_0} * \frac{q_1 * q_2}{r^2} * \vec{e_r}$

[3]Hooksches Gesetz: $\vec{F} = -D * \vec{s}$

berechnen, wobei die Konstante $c_s$ die Federkraft und $l_{uv}$ die natürliche Länge der Feder angibt.
Die Kraft $f_0$ in $x$-Richtung lautet:

$$F_{0_x}(u) = \frac{c_c}{(p_{v_x} - p_{u_x})^2 + (p_{v_y} - p_{u_y})^2} * \frac{p_{v_x} - p_{u_x}}{\sqrt{(p_{v_x} - p_{u_x})^2 + (p_{v_y} - p_{u_y})^2}}$$

Analog lassen sich die Werte für $F_{0_y}$, $F_{1_x}$ und $F_{1_y}$ berechnen.

Die Kraft, die auf einen Knoten $u \in V$ einwirkt, läßt sich folgendermaßen bestimmen:

$$F(u) = \sum_{(u,v) \in V \times V} f_0(u,v) + \sum_{(u,v) \in E} f_1(u,v)$$

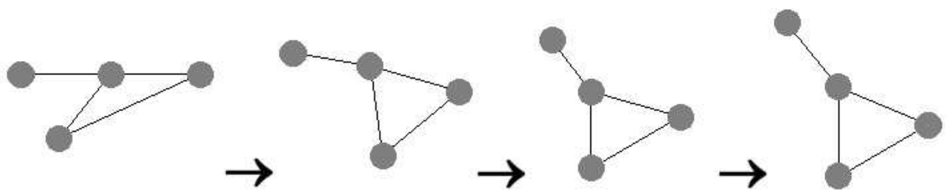
# 3 Berechnung des Layouts

Auf Knoten, die noch nicht im “Kräftegleichgewicht“ sind, wirkt eine Kraft. Um das System zu “entspannen“, werden die Knoten iterativ in Richtung der auf sie einwirkenden Kraft verschoben. Zum Zeitpunkt $t$ wirkt auf einen Knoten $v$ die Kraft $F_t(v)$. Nach Berechnung aller Kräfte $F_t(v)$ für alle $v \in V$ zum Zeitpunkt $t$, wird jeder Knoten $v$ um $\delta * F_t(v)$ verschoben. Die Konstante $\delta$ $(0 < \delta < 1)$ verhindert, daß Knoten übermäßig weit verschoben werden.

**Algorithmus: Spring-Embedder**

WHILE (...)
  FOR $v \in V$ DO
    $F_v = \sum_{(v,w) \in V \times V} f_0 + \sum_{(v,w) \in E} f_1$
  OD

  FOR $v \in V$ DO
    $p_v \leftarrow p_v + \delta * F_v$
  OD
OD



Ablauf des Spring-Embedders

### 3.1 Abbruchbedingung

Die einfachste Möglichkeit ist es, nach einer bestimmten Anzahl von Iterationen abzubrechen.

Eine weitere Möglichkeit ist es, die globale Kraft zu einem Zeitpunkt $t$ im System zu bestimmen. Diese kann folgendermaßen berechnet werden:

$$f_G = \sum_{v \in V} |f_t(v)|$$

Falls $f_G$ einen bestimmten Wert unterschreitet (im Idealfall $f_G = 0$), kann mit der Berechnung abgebrochen werden.

## 4 Beschleunigung der Berechnungen

### 4.1 Vermeiden der Berechnung $\sqrt{x}$

Durch Modifikation der Berechnungen von $f_0$ und $f_1$ kann die Rechen-Operation $\sqrt{x}$ vermieden werden:

$$\begin{aligned} f_0'(p_u, p_v) &= \frac{c_c}{||p_v - p_u||^2} * \frac{(p_v - p_u)}{||p_v - p_u||^2} \\ &= \frac{c_c * (p_v - p_u)}{||p_v - p_u||^4} \end{aligned}$$

$$f_1'(p_u, p_v) = c_s(||p_v - p_u||^2 - l_{uv}) * \frac{(p_v - p_u)}{||p_v - p_u||^2}$$

### 4.2 Clipping

Zur Berechnung der Kraft $F(u)$ eines Knotens $u \in V$ wird $\sum_{(u,v) \in V \times V} f_0(u, v)$ bestimmt. Diese Berechnung wird pro Iteration $n * (n-1) = O(n^2)$ durchgeführt. Knoten, die weit von einander entfernt sind, stoßen sich nur sehr gering ab. Aus diesem Grund kann man bei der Berechnung der abstoßenden Kraft diese weit entfernten Knoten vernachlässigen. Die Kraft $f_0(u, v)$ wird nur für Knoten $v$ berechnet, deren Abstand zu $u$ ($||p_v - p_u||^2$) $< d^2$ ist.

## 5 Weiteres

### 5.1 Variationen

Der obige Algorithmus kann auf unterschiedliche Weisen variiert werden. Seien $c_V(v) : V \to \mathbb{R}$ und $c_E(e) : E \to \mathbb{R}$ Funktionen, die den Knoten bzw. Kanten Gewichte zuordnen. Anstatt die Werte $c_c$ und $c_s$ als Konstanten zu verwenden, wird in $f_0(u, v)$ $c_c$ durch $c_{c_u} * c_{c_v}$ ersetzt, wobei $c_{c_v} = c_V(v)$ und in $f_1(u, v)$ wird $c_s$ durch $c_E(\{u, v\})$ ersetzt.

### 5.2 Probleme

Der gegebene Algorithmus funktioniert nur auf zusammenhängenden Graphen $G$. Besteht der Graph $G$ aus mehreren Zusammenhangs-Komponenten $Z_1, ..., Z_n$, so driften die einzelnen Zusammenhangs-Komponenten $Z_i$ in jedem Iterationsschritt auseinander und es stellt sich kein Kräftegleichgewicht ein.

Dieses Problem läßt sich auf zwei unterschiedliche Arten lösen:

- Man fügt einen weiteren Knoten $w$ ein, und verbindet diesen entweder mit *jedem* Knoten $v \in G$ oder mit *einem* Knoten $v$ in jeder Zusammenhangs-Komponente $Z_i$. Nach der Berechnung des Layouts entfernt man wieder diesen Knoten $w$ und die entsprechenden Kanten.

- Die Ränder des Zeichenbereiches erhalten eine Ladung, so daß die Knoten abgestoßen werden.

## Literatur

[1] Ulrik Brandes. Drawing on Physical Analogies. Drawing Graphs. LNCS 2025: 71–86, 2001.

[2] Guiseppe di Battista and Peter Eades and Roberto Tamassia and Ioannis G. Tollis. Graph Drawing. Algorithms for the Visualizaition of Graphs. 303–325.